Equivalence of Learning Algorithms

Julien Audiffren, Hachem Kadri

Equivalence of Learning Algorithms

Julien Audiffren[1] and Hachem Kadri[2]

[1]CMLA, ENS Cachan,Cachan,France
[2]QARMA, Aix-Marseille Universit, CNRS, LIF, Marseille, France

Abstract

The purpose of this paper is to introduce a concept of equivalence between machine learning algorithms. We define two notions of algorithmic equivalence, namely, weak and strong equivalence. These notions are of paramount importance for identifying when learning properties from one learning algorithm can be transferred to another. Using regularized kernel machines as a case study, we illustrate the importance of the introduced equivalence concept by analyzing the relation between kernel ridge regression (KRR) and m-power regularized least squares regression (M-RLSR) algorithms.

1 Introduction

Equivalence is a fundamental concept that defines a relationship between two objects and allows the inference of properties of one object from the properties of the other. In the field of machine learning, several theoretical concepts have been proposed in order to estimate the accuracy of learning algorithms, among which complexity of the hypothesis set [1, 2, 3], stability [4] and robustness [5], but little can be said concerning how these learning properties can be moved from one learning algorithm to another one. This raises the question of how equivalence between two learning algorithms might be defined such that some learning characteristics could be transferred between them.

When are two learning algorithms equivalent? More precisely, given $\{Z, \lambda_i, \mathcal{A}_i\}$, $i = 1, 2$, where Z is a training set, $\mathcal{A}_i$ is a learning algorithm that constructs to every training set Z a decision function $f^i_{Z,\lambda}$ and λ_i is a tuning parameter that balances the trade-off between fitness of $f^i_{Z,\lambda}$ to the data Z and smoothness of $f^i_{Z,\lambda}$, under what conditions is $\mathcal{A}_1$ equivalent to $\mathcal{A}_2$? Many learning algorithms can be formulated as an optimization problem. In this case, it is often thought that two learning algorithms are equivalent if their associated optimization problems are. One purpose of this paper is to point out that showing that two optimization problems are equivalent is not adequate evidence that the underlying learning algorithms are exchangeable with each other or even share some learning properties. Indeed, restricting the analysis of equivalence between learning algorithms to that of their optimization problems tends to omit details and ignore steps forming the whole learning mechanism such as the selection of the tuning parameter λ or the change of the learning set Z. The notion of equivalence between learning algorithms needs thus to be clearly defined.

The concept of *algorithmic equivalence* in machine learning has never been properly defined; the degree of difference between an optimization problem and its associated learning algorithm is not defined anywhere, nor has an exact definition. In this work, we rigorously define two notions of equivalence for learning algorithms and show how to use these notions to transfer stability property from a learning algorithm to another. Algorithmic stability is of particular interest since it provides a sufficient condition for a learning algorithm to be consistent and generalizing [6]. The first notion of equivalence we define, called weak-equivalence, is related in some way to the equivalence between

the associated minimization problems. By weak equivalence, we would like to emphasize here that this equivalence holds only when the algorithms are evaluated on a given training set Z. This matches in some manner the equivalence between the optimization problems since the objective function to minimize is evaluated only for the set Z of training examples $(x_i, y_i)_{i=1}^n$, which is fixed in advance. The second notion is stronger, in the sense that two learning algorithms are strongly equivalent when their equivalence does not depend on the training set Z.

As a case study, we consider regularized kernel methods which are learning algorithms with a regularization over a reproducing kernel Hilbert Space (RKHS) of functions. In particular, we study the regularized least squares regression problem when the RKHS regularization is raised to the power of m [7, 8], where m is a variable real exponent, and design an efficient algorithm for computing the solution, called M-RLSR (m-power regularized least squares regression). Using our algorithmic equivalence concept, we analyze the relation between M-RLSR and kernel ridge regression (KRR) algorithms.

In this paper, we make the following contributions: **1)** we formalize the concept of equivalence between two learning algorithms and define two notions of algorithmic equivalence, namely, weak and strong equivalence (Section 3). **2)** We show that the weak equivalence is not sufficient to allow the transfer of learning properties, such as stability, while strong equivalence is. Moreover, we provide sufficient assumptions under which the transfer of stability still holds even in the weak equivalence case (Section 4). **3)** As a case study, we consider the equivalence between KRR and M-RLSR. More precisely, we derive a semi-analytic solution to the M-RLSR optimization problem, we design an efficient algorithm for computing it, and we show that M-RLSR and KRR algorithms are weakly and not strongly equivalent (Section 5).

2 Notations and Backround

In the following, $\mathcal{X}$ will denote the input space, $\mathcal{Y}$ the output space, $\mathcal{Z}^n = (\mathcal{X} \times \mathcal{Y})^n$, $\mathcal{Z} = \cup_{n \geq 1} \mathcal{Z}^n$, $Z \in \mathcal{Z}$ a training set, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ the Banach space of hypotheses (for instance a separable reproducing kernel Hilbert space (RKHS)). Here and throughout the paper we use both notations $|Z|$ and $\#(Z)$ to denote the cardinal of the set Z.

Following the work of Bousquet and Elisseeff [4], a learning algorithm is defined as a mapping that takes a learning set made of input-output pairs and produces a function f that relates inputs to the corresponding outputs. For a large class of learning algorithms, the function f is obtained by solving an optimization problem. So before talking about *algorithmic equivalence*, it is helpful to discuss equivalence between optimization problems. A mathematical optimization problem [9] has the form

$$\begin{aligned} &\textit{minimize} \quad l(f) \\ &\textit{subject to} \quad l_i(f) \leq b_i, \quad i = 1, \ldots, q. \end{aligned}$$

Here f is the optimization variable of the problem, the function $l : \mathcal{H} \to \mathbb{R}$ is the objective function, the functions l_i are the (inequality) constraint functions, and the constants b_i are the limits for the constraints. A function f^* is called optimal, or solution of the problem, if it has the smallest objective value among all functions that satisfy the constraints. We consider here only strictly convex objective and constraint functions, so that the minimization problem have an unique solution and f^* is well defined. Two optimization problems are equivalent if both provide the same optimal solution.

The machine learning literature contains studies showing equivalence between learning algorithm (see, e.g., [10, 11, 12]); however, most of them have focused only on the equivalence that may occur between the associated optimization problems. In this sense, an equivalence between two optimization problems offers a way to relate the associated learning algorithms. However, this is not sufficient to decide whether the optimization equivalence allows to transfer theoretical properties from one learning algorithm to the other. A work that have indirectly supported this view

is that of Rifkin [13], who studied learning algorithms related to Thikonov and Ivanov regularized optimization. In [13, Chapter 5], it was shown that even though these optimization problems are equivalent, i.e., they give the same optimal solution, the associated learning algorithms have not the same stability properties. From this point of view, equivalence between minimization problems does not imply that the underlying algorithms share the same learning characteristics. This consideration is closely related to our goal of identifying when properties of a learning algorithm can be moved from one to another. The concept of *algorithmic equivalence* emerges in response to this question.

3 Weak and Strong Equivalence Between Learning Algorithms

In this section, we provide a rigorous definition of the concept of equivalence between machine learning algorithms. The idea here is to extend the notion of equivalence of optimization problems to learning algorithms. We first start by recalling the definition of a learning algorithm as given by Bousquet and Elisseeff [4], and for simplicity we restrict ourselves to learning algorithms associated to strictly convex optimization problems.

Definition 3.1 *(Learning Algorithm). A learning algorithm $\mathcal{A}$ is a function $\mathcal{A} : \mathcal{Z} \to \mathcal{H}$ which maps learning set Z onto a function $A(Z)$, such that*

$$\mathcal{A}(Z) = \arg\min_{g \in \mathcal{H}} R(Z, g), \tag{1}$$

where $R(Z, \cdot)$ is a strictly convex objective function.

Since we consider only strictly convex objective functions, the minimization problem has an unique solution and (1) is well defined. From this definition, the following definition of equivalence between algorithm naturally follows.

Definition 3.2 *(Equivalence). Let Z be a training set. Two algorithms $\mathcal{A}$ and $\mathcal{B}$ are equivalent on Z if and only if $\mathcal{A}(Z) = \mathcal{B}(Z)$.*

In other words, let $\mathcal{A}$ (resp. $\mathcal{B}$) be a learning algorithm associated to the optimization problem R (resp. S), then $\mathcal{A}$ and $\mathcal{B}$ are equivalent on Z if and only if the optimal solution of $R(Z, \cdot)$ is the optimal solution of $S(Z, \cdot)$. It is important to point out that the optimal solutions of R and S are computed for a set Z, and even though they are equal on Z, there is no guarantee that this remains true if Z varies. This means that the two algorithms $\mathcal{A}$ and $\mathcal{B}$ provide the same output with the set Z, but this may not be necessarily the case with another set Z'.

In this paper, we pay special attention to regularized learning algorithms. These algorithms depend on a regularization parameter that plays a crucial role in controlling the trade-off between overfitting and underfitting. It is important to note that many widely used regularized learning algorithms are families of learning algorithms. Indeed, each value of the regularization parameter λ defines a different minimization problem. As an example, we consider Kernel ridge regression [14] which is defined as follows

$$\arg\min_{f \in \mathcal{H}} \frac{1}{|Z|} \sum_{(x,y) \in Z} (y - f(x))^2 + \lambda \|f\|^2_{\mathcal{H}}. \tag{2}$$

In the following, we denote by $\mathbf{A}(\cdot)$ a regularized learning algorithm indexed by a regularization parameter in $\mathbb{R}^*_+$. In other words, $\forall \lambda \in \mathbb{R}^*_+$, $\mathcal{A}(\cdot) \doteq \mathbf{A}(\lambda)(\cdot) = \mathbf{A}(\lambda, \cdot)$ defines a learning algorithm as in definition 3.1. Note that all the results we will state can be naturally extended to a larger class of family of learning algorithms.

Equivalence of regularized learning algorithms. We now define the notion of weak equivalence between regularized learning algorithms as an extension of definition 3.2. To illustrate this equivalence, we provide some basic examples in this section and an in-depth case study in Section 5.

Definition 3.3 *(Weak Equivalence). Let* $\mathbf{A}(\cdot)$ *and* $\mathbf{B}(\cdot)$ *two regularized learning algorithms on* $\mathbb{R}_+^* \times \mathcal{Z}$. *Then* $\mathbf{A}(\cdot)$ *and* $\mathbf{B}(\cdot)$ *are said to be weakly equivalent if and only if* $\exists \Phi_{\mathbf{A}\to\mathbf{B}} : \mathbb{R}_+^* \times \mathcal{Z} \mapsto \mathbb{R}_+^*$ *such that*

1. $\forall Z \in \mathcal{Z}$, $\Phi_{\mathbf{A}\to\mathbf{B}}(\cdot, Z)$ *is a bijection from* $\mathbb{R}_+^*$ *into* $\mathbb{R}_+^*$,
2. $\forall Z \subset \mathcal{Z}, \quad \forall \lambda \in \mathbb{R}_+^*$, $\mathbf{A}(\lambda)$ *and* $\mathbf{B}(\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z))$ *are equivalent on* Z.

In the particular case where $\Phi_{\mathbf{A}\to\mathbf{B}}$ does not depend on Z, this assertion becomes much stronger than the weak equivalence, and will be referred as strong equivalence.

Definition 3.4 *(Strong Equivalence). Let* $\mathbf{A}(\cdot)$ *and* $\mathbf{B}(\cdot)$ *two regularized learning algorithms on* $\mathbb{R}_+^* \times \mathcal{Z}$. *Then* $\mathbf{A}(\cdot)$ *and* $\mathbf{B}(\cdot)$ *are said to be strongly equivalent if and only if it exists* $\Phi_{\mathbf{A}\to\mathbf{B}}$, *a bijection from* $\mathbb{R}_+^*$ *into* $\mathbb{R}_+^*$ *such that* $\mathbf{A}(\cdot) = \mathbf{B}(\Phi_{\mathbf{A}\to\mathbf{B}}(\cdot))$ *where the equality is among functions from* $\mathbb{R}_+^*$ *into* $\mathcal{H}^{\mathcal{Z}}$.

This notion of weak equivalence is frequently encountered in machine learning algorithms. For instance, it naturally occurs when using Lagrangian duality and when transiting from Ivanov's to Thikonov's method [13, Chapter 5]. Note that weak and strong equivalence between two learning algorithms have some immediate implications for interpreting their regularization paths (see the supplementary material for more details). The natural question which now arises is whether by knowing some learning properties of $\mathbf{A}$ and the weak equivalence of $\mathbf{A}$ and $\mathbf{B}$ it is possible to deduce learning properties for $\mathbf{B}$. This question is studied in the next section.

4 Consequences of Equivalence Between Learning Algorithms

We now study the consequences of the algorithmic equivalences defined in the previous section. In particular we investigate wether these notions of equivalence allow the transfer of learning properties from one learning algorithm to another. We first begin by the following proposition which presents a main of the weak equivalence.

Proposition 4.1 *Let* $\mathbf{A}(\lambda)$ *and* $\mathbf{B}(\lambda)$ *two weakly equivalent regularized learning algorithms. Then*

$$\forall Z \subset \mathcal{X} \times \mathcal{Y}, \quad \inf_{\lambda\in\mathbb{R}} \mathbf{A}(\lambda)(Z) = \inf_{\lambda\in\mathbb{R}} \mathbf{B}(\lambda)(Z).$$

Proof : This Proposition directly follows from Definition 3.3.

Proposition 4.1 means that the optimal solutions given by two weakly equivalent (regularized) learning algorithms are the same. However, without further assumptions, weak equivalence is of little consequence to the transfer of learning properties from one to the other, such as stability, consistency or generalization bounds. Indeed, these properties are defined for a varying training set either by altering it (such as in stability) or by making it increasingly large (such as in consistency). To illustrate this idea, we will address in particular the question whether weakly equivalence allows or not the transfer of stability.

Transfer of stability. In the following we choose to focus on uniform stability which is an important property of a learning rule that allows to get bounds on the generalization performance of learning algorithms. Following [4], the uniform stability of a regularized learning algorithm is defined as follows.

Notation. $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ denotes a loss function on $\mathcal{Y}$, and $\forall Z \in \mathcal{Z}$, $\forall 1 \leq i \leq |Z|$, Z^i denotes the set Z minus its i-th element.

Definition 4.2 *(Uniform stability). Let $\beta : \mathbb{N}^+ \times \mathbb{R}_+^* \mapsto \mathbb{R}_+$ be such that $\forall \lambda > 0, \lim_{n\to\infty} \beta(n,\lambda) = 0$. A regularized learning algorithms $\mathbf{A}$ is said to be β- uniformly stable with respect to ℓ if*

$$\forall \lambda \in \mathbb{R}_+^*, \quad \forall Z \in \mathcal{Z} \quad \forall 1 \leq i \leq n \quad \forall (x,y) \in \mathcal{X} \times \mathcal{Y} \quad |\ell(y, \mathbf{A}(\lambda, Z)) - \ell(y, \mathbf{A}(\lambda, Z^i))| \leq \beta(|Z|, \lambda).$$

We now give an example to show that weak equivalence is not a sufficient condition for the transfer of uniform stability.

Example: Let $\mathbf{A}(\cdot)$ denotes the KRR as defined by (2) and $\mathbf{B}(\cdot)$ denotes a modified KRR where the regularization term is $\lambda/|Z|$ instead of λ. $\mathbf{A}$ and $\mathbf{B}$ are weakly equivalent with $\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z) = \lambda|Z|$. Under some widely used hypotheses on the kernel and on the output random variable Y [4], $\mathcal{A}$ is known to be β uniformly stable with $\beta(n,\lambda) = C_1(1 + C_2/\sqrt{\lambda})/n\lambda$ where C_1 and C_2 are constants, and n is the size of the training set Z (see e.g. [4] or [15] for more details). Similarly, it is easy to see that $\mathbf{B}$ satisfies the same property but with $\beta(n,\lambda) = C_1(1 + C_2/\sqrt{\lambda n})/\lambda$, which no longer tends to ∞ as n increases. Indeed, the regularization term in the learning algorithm $\mathbf{B}(\lambda)$ decreases as $|Z|$ increases, and this leads to a decrease of the stability of the learning algorithm.

It is important to note that if $\mathbf{A}$ and $\mathbf{B}$ are strongly equivalent, then unlike in the weak equivalence case, many properties of $\mathbf{B}$ are transferred to $\mathbf{A}$. The following Lemma illustrates this idea in the case of stability.

Lemma 4.3 *If $\mathbf{A}$ and $\mathbf{B}$ are strongly equivalent and if $\mathbf{B}$ is $\beta(\cdot,\cdot)$ uniformly stable, then $\mathbf{A}$ is $\beta(\cdot, \Phi_{\mathbf{A}\to\mathbf{B}}(\cdot))$ uniformly stable.*

Proof : Let $(x,y) \in \mathcal{X} \times \mathcal{Y}$, $Z \in \mathcal{Z}$ and $\lambda \in \mathbb{R}_+^*$. It is easy to see that

$$\begin{aligned} |\ell(y, \mathbf{A}(\lambda,Z)(x)) - \ell\big(y, \mathbf{A}(\lambda,Z^i)(x)\big)| &= |\ell(y, \mathbf{B}(\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda), Z)(x)) - \ell\big(y, \mathbf{B}(\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda), Z^i)(x)\big)|, \\ &\leq \beta(|Z|, \Phi_{\mathbf{A}\to\mathbf{B}}(\lambda)). \end{aligned}$$

□

We have shown that, contrarily to strong equivalence, weak equivalence is not sufficient to ensure the transfer of learning properties such as uniform stability.

In the following, we introduce two additional assumptions which are a sufficient condition for the transfer of the uniform stability under the weak equivalence. In order to clearly express these assumptions, we first introduce a metric on training set, that is to say a metric *on unordered sequences of different lengths*. To the best of our knowledge, this is a new metric, which allows to easily express learning properties such as stability. We will refer to this metric as the generalized Hamming metric[1] (see e.g. [16] for more details on the usual Hamming metric).

Definition 4.4 *Let $n > 0$, $Z^1 = \{z_1^1, \ldots, z_n^1\}$ and $Z^2 = \{z_1^2, \ldots, z_n^2\}$. Let $\Sigma(n)$ denotes the set of all the permutations of $\{1, \ldots, n\}$ and $\underline{H}$ denotes the usual Hamming metric on sequences. $\forall \sigma \in \Sigma(n)$, we denote by Z_σ^1 the sequence of n elements, whose i-th element is $z_\sigma^1(i)$. We define*

- $G_n : (\mathcal{X} \times \mathcal{Y})^n \times (\mathcal{X} \times \mathcal{Y})^n \mapsto \mathbb{R}^+$, *such that* $G_n(Z^1, Z^2) = \min_{\sigma\in\Sigma(n)} \underline{H}(Z_\sigma^1, Z^2)$,
- $\mathbf{H} : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}^+$, *such that*

$$\mathbf{H}(Z_1, Z_2) = \begin{cases} \#(Z_1) - \#(Z_2) + \min\limits_{Z\subset Z_1, \#(Z)=\#(Z_2)} G_{\#(Z_2)}(Z, Z_2) & \text{if } \#(Z_1) \geq \#(Z_2), \\ \#(Z_2) - \#(Z_1) + \min\limits_{Z\subset Z_2, \#(Z)=\#(Z_1)} G_{\#(Z_1)}(Z, Z_1) & \text{otherwise.} \end{cases} \tag{3}$$

The idea of this metric is to consider the number of deletion (i.e. removing an element), insertion (adding an element) and change (changing the value of one element) that allows to move from one training set to another (permutations of two elements among a training set are free). The following proposition proves that $\mathbf{H}$ is indeed a metric on $\mathcal{Z}$.

[1] Note that the Hamming metric and the generalized Hamming metric do not coincide on the set of ordered sequences.

Proposition 4.5 *The function* $\mathbf{H} : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}$ *defined in definition 4.4 is a metric over* $\mathcal{Z}$.

PROOF : see the supplementary material. □

Remark 4.6 *The generalized Hamming metric can be used to reformulate the notion of stability. For instance,* $\mathcal{A}$ *is* β *uniformly stable if and only if* $\mathcal{A}$ *is* β *lipschitz with respect to the metric* $\mathbf{H}$.

With the help of the metric $\mathbf{H}$, we now introduce two assumptions on the regularity of the functions $\Phi_{\mathbf{A}\to\mathbf{B}}$ and $\mathbf{A}$.

Assumption 1 $\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, \cdot)$ *is* C *Lipschitz decreasing with respect to* $\mathbf{H}$, *i.e.* $\exists c > 0$ *and* $C : \mathbb{N} \mapsto \mathbb{R}$, *decreasing,* $\lim_{n\to\infty} C(n) = 0$, *such that*

1. $\forall \lambda \in \mathbb{R}_+^*, \quad \forall Z_1, Z_2 \in \mathcal{Z}, \quad |\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z_1) - \Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z_2)| \leq c\mathbf{H}(Z_1, Z_2)$,
2. $\forall \lambda \in \mathbb{R}_+^*, \quad \forall \in \mathcal{Z}, \quad \forall 1 \leq i \leq |Z|, \quad |\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z) - \Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z^i)| \leq C(|Z|)$.

Assumption 2 *Let* $\gamma > 0$. $\mathbf{A}$ *is* γ *Lipschitz with respect to its first variable, i.e.* $\forall Z \in \mathcal{Z}$, $\forall \lambda_1, \lambda_2 \in \mathbb{R}_+^*$, $\|\mathbf{A}(\lambda_1)(Z) - \mathbf{A}(\lambda_2)(Z)\|_{\mathcal{H}} \leq \gamma|\lambda_1 - \lambda_2|$.

These two assumptions are a sufficient condition to the transfer of stability in the weak equivalence case, as shown in the following Proposition.

Proposition 4.7 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two weakly equivalent regularized learning algorithms satisfying Assumptions 1 and 2. Moreover, let* β *be as in Definition 4.2 and locally Lipschitz with respect to its second variable. Suppose that* $\exists \kappa > 0$ *such that* $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}$, $\|f(x)\|_{\mathcal{Y}} \leq \kappa \|f\|_{\mathcal{H}}$. *Then:*

If $\mathbf{B}$ *is* β *uniformly stable, then* $\mathbf{A}$ *is* β' *uniformly stable with* $\forall \lambda \in \mathbb{R}_+^*$, $\beta'(\cdot, \lambda) = O(\beta(\cdot, \lambda) + C(\cdot))$.

PROOF : Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $Z \in \mathcal{Z}$, $n = |Z|$ and $\lambda \in \mathbb{R}_+^*$. First note that since ℓ is σ-admissible, by using $\lambda' = \Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z)$ and $\lambda'' = \Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z^i)$,

$$
\begin{aligned}
&|\ell\left(y, \mathbf{A}(\lambda, Z)(x)\right) - \ell\left(y, \mathbf{A}(\lambda, Z^i)(x)\right)| = |\ell\left(y, \mathbf{B}(\lambda', Z)(x)\right) - \ell\left(y, \mathbf{B}(\lambda'', Z^i)(x)\right)|, \\
&\leq |\ell\left(y, \mathbf{B}(\lambda', Z)(x)\right) - \ell\left(y, \mathbf{B}(\lambda', Z^i)(x)\right)| + |\ell\left(y, \mathbf{B}(\lambda', Z^i)(x)\right) - \ell\left(y, \mathbf{B}(\lambda'', Z^i)(x)\right)|, \\
&\leq \beta(n, \lambda') + \sigma\kappa\|\mathbf{B}(\lambda', Z^i) - \mathbf{B}(\lambda'', Z^i)\|_{\mathcal{H}}, \leq \beta(n) + \sigma\kappa\gamma|\lambda' - \lambda''|, \\
&\leq \beta(n, \lambda) + \delta C(n) + \sigma\kappa\gamma C(n),
\end{aligned}
$$

where in the last line we used the fact that β is locally Lipschitz with respect to λ, hence $\exists \delta(\lambda, |\lambda - \lambda'|) > 0$ such that $\beta(n, \lambda') \leq \beta(n, \lambda) + \delta|\lambda - \lambda'|$. Now, since $\Phi_{\mathbf{A}\to\mathbf{B}}$ is C-Lipschitz decreasing, $|\lambda' - \lambda| \to_{n\to\infty} 0$, hence the conclusion. □

Remark 4.8 *Proposition 4.7 can be extended to some non* σ*-admissible loss such as the the square loss by using the same ideas as in [4] and [15].*

The next section is devoted to present an in-depth case study of weak equivalence. It introduces a new regularized learning algorithm, M-RLSR, and studies the equivalence between KRR and M-RLSR.

5 Case Study: M-RLSR

Notation. In this section, $m > 0$ is a real number, $\mathcal{X}$ a Hilbert space, $\mathcal{Y} = \mathbb{R}$, $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ a separable reproducing kernel Hilbert space (RKHS), and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ its positive definite kernel. For all set of n elements of $\mathcal{X} \times \mathbb{R}$, we denote by $Z = \{(x_1, y_1), .., (x_n, y_n)\}$ the training set, and by K the

Algorithm 1 M-Power RLS Regression Algorithm (M-RLSR)

Input: training data $Z = \{(x_1, y_1), .., (x_n, y_n)\}$, parameter $\lambda \in \mathbb{R}_+^*$, exponent $m \in \mathbb{R}_+^*$

1. **Kernel matrix:** Compute the Gram matrix, $K = \big(k(x_i, x_j)\big)_{1 \leq i,j \leq n}$
2. **Matrix diagonalization:** Diagonalize K in an orthonormal basis
$$K = QDQ^\top \quad ; \quad d_i = D_{ii} \ , \ \forall 1 \leq i \leq n$$
3. **Change of basis:** Perform a basis transformation
$$Y = Q^\top Y \quad ; \quad y_i = Y_i \ , \ \forall 1 \leq i \leq n$$
4. **Root-finding:** Find the root C_0 of the function F defined in (10)
5. **Solution:** Compute α from (9) and reconstruct the weights
$$(\alpha_i)_{1 \leq i \leq n} = \frac{2y_i}{2d_i + \lambda m n C_0} \quad \text{and} \quad \alpha = Q\alpha$$

Gram matrix associated to k for Z with $(K_Z)_{i,j} = k(x_i, x_j)$. Finally, let $Y = (y_1, ..., y_n)^\top$ be the output vector.

The algorithm we investigate here combines a least squares regression with an RKHS regularization term raised to the power of m. Formally, we would like to solve the following optimization problem:

$$f_Z = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^m, \tag{4}$$

where m is a suitable chosen exponent. Note that the classical kernel ridge regression (KRR) algorithm [14] is recovered for $m = 2$. This problem has been studied from a theoretical point of view (see [7, 8]), and in this section we propose a practical way to solve it. The problem (8) is well posed for $m > 1$. We now introduce a novel m-power RLS regression algorithm, generalizing the kernel ridge regression algorithm to an arbitrary regularization exponent.

5.1 M-RLSR Algorithm

It is worth recalling that the minimization problem (8) with $m = 2$ becomes a standard kernel ridge regression, which has an explicit analytic solution. In the same spirit, the main idea of our algorithm is to derive analytically from (8) a reduced one-dimensional problem on which we apply a root-finding algorithm.

By applying the generalized Representer Theorem from [17], we obtain that the solution of (8) can be written as $f_Z = \sum_{i=1}^n \alpha_i k(., x_i)$, with $\alpha_i \in \mathbb{R}$. The following theorem gives an efficient way to compute the vector $\alpha = (\alpha_1, ..., \alpha_n)^\top$.

Theorem 1 *Let Q an orthonormal matrix and D a diagonal matrix such that $K = QDQ^\top$. Let y'_i be the coordinates of $Q^T Y$, $(d_i)_{1 \leq i \leq n}$ the elements of the diagonal of D, $C_0 \in \mathbb{R}_+$ and $m > 1$. Then the vector $\alpha = Q\alpha'$ with*

$$\alpha'_i = \frac{2y'_i}{2d_i + \lambda m n C_0} \ , \ \forall 1 \leq i \leq n \ , \tag{5}$$

is the solution of (8) if and only if C_0 is the root of the function $F : \mathbb{R}_+ \to \mathbb{R}$ defined by

$$F(C) = \Big(\sum_{i=1}^{n} \frac{4 d_i y'^2_i}{(2d_i + \lambda m n C)^2} \Big)^{m/2-1} - C. \tag{6}$$

PROOF : The proof of Theorem 2 can be found in the supplementary material. □

It is important to note that for $m > 1$, F has a unique root C_0 and that since F is a function from $\mathbb{R}$ to $\mathbb{R}$, computing C_0 using a root-finding algorithm, e.g. Newton's method, is a fast and accurate procedure. Our algorithm uses these results to provide an efficient solution to regularized least squares regression with a variable regularization exponent m (see Algorithm 1).

5.2 Equivalence Between M-RLSR and KRR

Here we will show that M-RLSR and KRR are only weakly equivalent but not strongly equivalent. The idea is that, when $m > 1$, the objective function of the M-RLSR optimization problem (8) is strictly convex, and then by Lagrangian duality it is equivalent to its unconstrained version. The weak equivalence is proved in the following proposition.

Proposition 5.1 *$\forall m > 1$, $\forall Z \in \mathcal{Z}, \exists F_{Z,m} : \mathbb{R}^+ \to \mathbb{R}^+$, bijective, such that $\forall \lambda > 0$, M-RLSR with regularization parameter λ and KRR with regularization parameter $\lambda_2 = F_{Z,m}(\lambda)$ are weakly equivalent. Moreover,*

$$\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z) = \frac{m\lambda}{2} C_0(Z, m, \lambda),$$

where $C_0(Z, m, \lambda)$ is the unique root of the function F defined in (10).

PROOF : For $m > 1$, the equivalence between constrained and unconstrained strictly convex optimization problems [18, Appendix A] implies that $\exists \Gamma_{m,Z,\lambda} > 0$ such that the minimization problem defined by (8) on Z it is equivalent to the following constrained problem:

$$\arg\min_{f\in\mathcal{H}} \frac{1}{|Z|} \sum_{(x,y)\in Z} (y - f(x))^2, \quad \text{s.t. } \|f\|_{\mathcal{H}}^m \leq \Gamma_{m,Z,\lambda}.$$

The constrain is equivalent to $\|f\|_{\mathcal{H}}^2 \leq \Gamma_{m,Z,\lambda}^{2/m}$, thus we deduce that $\exists \lambda_2(m, Z, \lambda) > 0$ such that (8) with regularization parameter λ is equivalent to

$$\arg\min_{f\in\mathcal{H}} \frac{1}{|Z|} \sum_{(x,y)\in Z} (y - f(x))^2 + \lambda_2(m, Z, \lambda)\|f\|_{\mathcal{H}}^2,$$

i.e., the KRR minimization problem with a regularization parameter $\lambda_2(m, Z, \lambda)$. Hence M-RLSR with λ is weakly equivalent to KRR with $\lambda_2(m, Z, \lambda)$. It is easy to see from (9) that the function $F_{Z,m}$ that maps λ to the corresponding λ_2 has the form $F_{Z,m}(\lambda) := \frac{m}{2} C_0(Z, m, \lambda)\lambda$. □

Since $\Phi_{\mathbf{A}\to\mathbf{B}}(\lambda, Z)$ heavily depends on Z, M-RLSR and KRR are not strongly equivalent but only weakly equivalent. Moreover, Assumptions 1 and 2 are not satisfied in this case, hence stability of M-RLSR cannot be deduced from that of KRR. A stability analysis of M-RLSR can be found in the supplementary material.

5.3 Experiments on Weak Equivalence

In this subsection, we conduct experiments on synthetic and real-world datasets to illustrate the fact that M-RLSR and KRR algorithms are only weakly equivalent but not strongly equivalent. We use the Concrete Compressive Strength (1030 instances, 9 attributes) real-world dataset extracted from the UCI repository[2]. Additionally, we also use a synthetic dataset (2000 instances, 10 attributes) described in [19]. In this dataset, inputs $(x_1, ..., x_{10})$ are generated independently and uniformly over $[0, 1]$ and outputs are computed from $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \mathcal{N}(0, 1)$.

[2] `http://archive.ics.uci.edu/ml/datasets`.

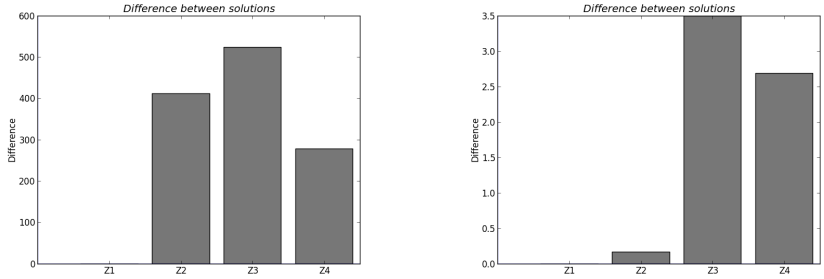

Figure 1: The norm of the difference between the optimal solutions given by M-RLSR and KRR on two datasets randomly split into 4 parts $Z_1, \ldots, Z_4$. While the difference on Z_1 is zero for the two algorithms since they are weakly-equivalent, they give different solution for Z_1, Z_2 and Z_3. (left) Concrete compressive strength dataset: $m = 1.5, \lambda = 1e-2$ (obtained by 10-fold cross validation) and $\lambda_2 = 5.6e-4$ (computed from Proposition 5.1). (right) Synthetic dataset: $m = 1.2, \lambda = 1e-5$ (obtained by 10-fold cross validation) and $\lambda_2 = 6.5e-7$ (computed from the Proposition 5.1).

We randomly split these datasets into 4 parts of equal size $Z_1, \ldots, Z_4$. Using Z_1, m is fixed and the regularization parameter λ is chosen by a 10-fold cross-validation for M-RLSR. Then the equivalent λ_2 for KRR is computed using Proposition 5.1. For each part $Z_i, 1 \leq i \leq 4$, we calculate the norm of the difference between the optimal solutions given by M-RLSR and KRR. The results are presented in Figure 1. The difference between the solutions of the two algorithms is equal to 0 on Z_1, but since both algorithms are only weakly equivalent, the difference is strictly positive on Z_2, Z_3, Z_4, showing that the algorithms are not strongly equivalent. Additional experiments regarding the M-RLSR and its algorithmic properties can be found in the supplementary material.

6 Conclusion

We have presented a novel way of theoretically analyzing and interpreting relations between machine learning algorithms, namely the concept of algorithmic equivalence. More precisely, we have proposed two notions of equivalence of learning algorithms, weak and strong equivalence, and we have shown how to use them to transfer learning properties, such as stability, from one learning algorithm to another.

Although this work has focused in particular on the transfer of stability using the concept of algorithmic equivalence, we believe that it can be extended to study the transfer of other algorithmic properties such as sparsity, robustness and generalization. Future work will also aim at further quantifying the equivalence relations introduced by providing efficient tools that can help to decide whether two learning algorithms are weakly or strongly equivalent.

References

[1] V. Vapnik and A.J. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1991.

[2] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[3] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[4] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[5] H. Xu and S. Mannor. Robustness and generalization. In *Proceedings of COLT*, pages 503–515, 2010.

[6] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability and stability in the general learning setting. In *Proceedings of COLT*, 2009.

[7] S. Mendelson and J. Neeman. Regularization in kernel learning. *The Annals of Statistics*, 38(1):526–565, 2010.

[8] I. Steinwart, D. Hush, C. Scovel, et al. Optimal rates for regularized least squares regression. In *COLT Proceedings*, 2009.

[9] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural computation*, 10(6):1455–1480, 1998.

[11] C. Rudin and R. E. Schapire. Margin-based ranking and an equivalence between AdaBoost and RankBoost. *Journal of Machine Learning Research*, 10:2193–2232, 2009.

[12] M. Jaggi. An equivalence between the lasso and support vector machines. In *International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines: Theory and Applications*, 2013.

[13] R. Rifkin. *Everything Old is New Again*. PhD thesis, Massachusetts Institute of Technology, 2002.

[14] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. Advances in Neural Information Processing Systems, 1998.

[15] J. Audiffren and H. Kadri. Stability of multi task kernel regression algorithms. ACML '14, 2014.

[16] R. Ash. *Information theory*. Dover Publications, 1965.

[17] F. Dinuzzo and B. Schölkopf. The representer theorem for Hilbert spaces: a necessary and sufficient condition. Advances in Neural Information Processing Systems, 2012.

[18] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953:997, 2011.

[19] I. W. Tsang, J. T. Kwok, and K. T. Lai. Core vector regression for very large regression problems. ICML, 2005.

A Weak Equivalence and Regularization Path

Let $Z \in \mathcal{Z}$ be a fixed training set, and $\mathbf{A}$ and $\mathbf{B}$ two weakly equivalent algorithm. By definition of the weak equivalence $\mathbf{A}(\cdot)(Z) = \mathbf{B}(\Phi_{\mathbf{A}\to\mathbf{B}}(\cdot, Z))(Z)$. This formulation highlights the consequence of the weak equivalence with the regularization path: the regularization path of $\mathbf{B}$ can be obtained from the the regularization path of $\mathbf{A}$ with the bijective transformation $\Phi_{\mathbf{A}\to\mathbf{B}}(\cdot, Z)$ of the variable λ. It is important to note that $\Phi_{\mathbf{A}\to\mathbf{B}}(\cdot, Z)$ depends on Z, i.e. *the relation between the regularization path of* $\mathbf{A}$ *and* $\mathbf{B}$ *depends on* Z. The same can be said for the error curves, but Proposition 4.1 ensure that they share the same minimum value (see figure 1).

B Proof of Proposition 4.6

Proof : It is easy to see that $\mathbf{H}$ is symmetric and $\mathbf{H}(Z_1, Z_2) \geq 0$ and is equal to 0 if and only if $Z_1 = Z_2$. Now, in order to prove the sub-additivity of $\mathbf{H}$, let Z_1, Z_2 and $Z_3 \in \mathcal{Z}$. Note that G_n counts the number of elements which differs between two unordered sequences, and thus is sub-additive.

We only write here the case $\#(Z_2) \geq \#(Z_1) \geq \#(Z_3)$, the other cases are done likewise. Let for $i = 1, 3$

$$Z_i^2 = \underset{Z\subset Z_2, \#(Z)=\#(Z_i)}{\arg\min} G_{\#(Z_i)}(Z, Z_i).$$

Without any loss of generality, suppose that $\#(Z_1^2) \geq \#(Z_3^2)$. Then,

$$\#(Z_2) \geq \#(Z_1^2) + G(\hat{Z}, Z_3^2), \text{ where } \hat{Z} = \underset{Z\subset Z_1^2, \#(Z)=\#(Z_3^2)}{\arg\min} G_{\#(Z_3^2)}(Z, Z_3^2). \tag{7}$$

$$\begin{aligned}
\mathbf{H}(Z_1, Z_2)+\mathbf{H}(Z_2, Z_3) =& \#(Z_2)-\#(Z_1)+G_{\#(Z_1)}(Z_1^2, Z_1)+\#(Z_2)-\#(Z_3) + G_{\#(Z_3)}(Z_3^2, Z_3) \\
\geq& \#(Z_2)-\#(Z_3)+G_{\#(Z_1)}(Z_1^2, Z_1)+G_{\#(Z_3)}(Z_3^2, Z_3) + G_{\#(Z_3)}(\hat{Z}, Z_3^2) \\
\geq& \#(Z_2) - \#(Z_3) + G_{\#(Z_1)}(Z_1^2, Z_1) + G_{\#(Z_3)}(\hat{Z}, Z_3) \\
\geq& \#(Z_2) - \#(Z_3) + \min_{Z\subset Z_1, \#(Z)=\#(Z_3)} G_{\#(Z_3)}(Z, Z_3) \\
\geq& \#(Z_1) - \#(Z_3) + \min_{Z\subset Z_1, \#(Z)=\#(Z_3)} G_{\#(Z_3)}(Z, Z_3) = \mathbf{H}(Z_1, Z_3)
\end{aligned}$$

where, in the second line we used (7), third line we used the sub-additivity of G, fourth line we used the fact that $\hat{Z} \subset Z_1^2$, and last line we used $\#(Z_2) \geq \#(Z_1) \geq \#(Z_3)$. □

C Proof of Theorem 1

Remember that we are trying to solve the following problem :

$$f_Z = \underset{f\in\mathcal{H}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|f\|_{\mathcal{H}}^m, \tag{8}$$

where m is a suitable chosen exponent.

For the convenience of the reader, let us rewrite the theorem we are going to prove

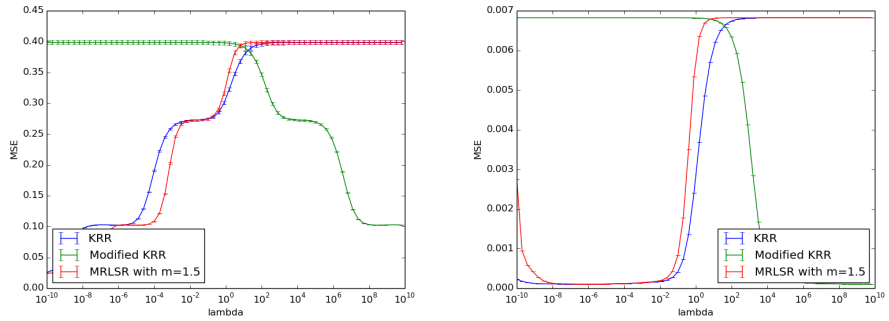

Figure 2: Error curves for KRR, modified KRR defined by (**??**) and M-RLSR with m=1.5 on the dataset Yatch hydrodynamic(left) and (right), both extracted from the UCI repository,

Theorem 2 *Let Q an orthonormal matrix and D a diagonal matrix such that $K = QDQ^\top$. Let y'_i be the coordinates of $Q^T Y$, $(d_i)_{1\leq i\leq n}$ the elements of the diagonal of D, $C_0 \in \mathbb{R}_+$ and $m > 1$. Then the vector $\alpha = Q\alpha'$ with*

$$\alpha'_i = \frac{2y'_i}{2d_i + \lambda mnC_0} \ , \ \forall 1 \leq i \leq n \ , \tag{9}$$

is the solution of (11) if and only if C_0 is the root of the function $F : \mathbb{R}_+ \to \mathbb{R}$ defined by

$$F(C) = \Big(\sum_{i=1}^{n} \frac{4d_i y'^2_i}{(2d_i + \lambda mnC)^2}\Big)^{m/2-1} - C. \tag{10}$$

Proof :

First notice that, the objective function to minimize is Gâteaux differentiable in every direction. Thus, since f_Z is a minimum, we have:

$$0 = \sum_{i=1}^{n} -2k(.,x_i)(y_i - f_Z(x_i)) + \lambda mn\|f_Z\|^{m-2}_{\mathcal{H}} f_Z,$$

i.e.,

$$f_Z = \sum_{i=1}^{n} 2k(.,x_i)\frac{y_i - f_Z(x_i)}{\lambda mn\|f_Z\|^{m-2}_{\mathcal{H}}}.$$

That is to say, f_Z can be written in the following form:

$$f_Z = \sum_{i=1}^{n} \alpha_i k(.,x_i), \tag{11}$$

with $\alpha_i \in \mathbb{R}$. Notice, that we have recovered exactly the form of the representer theorem, which can also be derived from a result due to Dinuzzo and Schölkop [17]. Now by combining (8) and (11), the initial problem becomes

$$\alpha = \underset{a\in\mathbb{R}^n}{\arg\min}(Y - Ka)^\top (Y - Ka) + n\lambda(a^\top Ka)^{m/2}, \tag{12}$$

where $\alpha = (\alpha_i)_{1\leq i\leq n}$ is the vector to determine. The following theorem gives an explicit formula for α that solves the optimization problem (12).

By computing the Gâteaux derivative of the objective function to minimize in (12), we obtain that α must verify

$$Y = K\alpha + \lambda \frac{mn}{2}(\alpha^\top K\alpha)^{m/2-1}\alpha.$$

Then, since K is symmetric and positive semidefinite, $\exists Q$ an orthonormal matrix (the matrix of the eigenvectors) and D a diagonal matrix with eigenvalues $(d_i)_{1\leq i\leq n} \geq 0$ such that $K = QDQ^\top$. Hence,

$$Y = QDQ^\top\alpha + \lambda \frac{mn}{2}((Q^\top\alpha)^\top D(Q^\top\alpha))^{m/2-1}\alpha$$
$$\Rightarrow Q^\top Y = DQ^\top\alpha + \lambda \frac{mn}{2}((Q^\top\alpha)^\top D(Q^\top\alpha))^{m/2-1}Q^\top\alpha.$$

Given this, one can define a new representation by changing the basis such that $Y' = Q^\top Y$ and $\alpha' = Q^\top\alpha$. We obtain

$$Y' = D\alpha' + \lambda \frac{mn}{2}(\alpha'^\top D\alpha')^{m/2-1}\alpha'.$$

Now if we write the previous equation for every coefficient of the vectors, we obtain that

$$\left\{ y'_i = d_i\alpha'_i + \lambda \frac{mn}{2}(\sum_{j=1}^{n} d_j\alpha'^2_j)^{m/2-1}\alpha'_i \quad , \quad \forall 1 \leq i \leq n. \right.$$

Note that $(\sum_{j=1}^{n} d_j\alpha'^2_j)^{m/2-1}$ is the same for every equation (i.e. it does not depend on i), so we can rewrite the system as follows, where $C \in \mathbb{R}$

$$\left\{ \begin{array}{l} C = \big(\sum_{j=1}^{n} d_j\alpha'^2_j\big)^{m/2-1} \\ \quad \text{and} \\ \alpha'_i = \dfrac{2y'_i}{2d_i + \lambda mnC} \quad , \quad \forall 1 \leq i \leq n. \end{array} \right. \tag{13}$$

which is well defined if $d_i + \lambda mnC \neq 0$, which is the case when $C > 0$. Since $C \geq 0$ by definition, the only possibly problematic case is $C = 0$, but this implies that $Y = 0$, which is a degenerated case. Now we just need to calculate C, which verifies:

$$C = \big(\sum_{i=1}^{n} d_i\alpha'^2_i\big)^{m/2-1} = \big(\sum_{i=1}^{n} \frac{4d_iy'^2_i}{(2d_i + \lambda mnC)^2}\big)^{m/2-1}.$$

Thus to obtain an explicit value for α', we need only to find a root of the function F defined as follows :

$$F(C) = \big(\sum_{i=1}^{n} \frac{4d_iy'^2_i}{(2d_i + \lambda mnC)^2}\big)^{m/2-1} - C.$$

We have proven that any solution of (12) can be written as a function of C_0, a root of F. But for $m > 1$, F is strictly concave, and $F(0) > 0$, hence it has at most one root in $\mathbb{R}_+$. Thus since $\lim_{C\to+\infty} F(C) = -\infty$, F has exactly one root, which proves Theorem 2. □

D Stability Analysis of M-RLSR

The notion of algorithmic stability, which is the behavior of a learning algorithm following a change of the training data, was used successfully by Bousquet and Elisseeff [4] to derive bounds on the generalization error of kernel-based learning algorithms. In this section, we extend the stability

results of [4] to cover the m-power RLSR algorithm. We show here that the algorithm is stable for $m \geq 2$.

In this section we denote by $\underline{X}$ and $\underline{Y}$ a pair of random variables following the unknown distribution D of the data, $\underline{X}$ representing the input and $\underline{Y}$ the output, by $Z^i = Z \setminus (x_i, y_i)$ the training set from which was removed the element i. Let $c(y, f, x) = (y - f(x))^2$ denotes the cost function used in the algorithm. For all $f \in \mathcal{H}$, let $R_e(f, Z) = 1/n \sum_{1 \leq i \leq n} c(y_i, f, x_i)$ be the empirical error and $R_r(f, Z) = R_e(f, Z) + \lambda \|f\|_{\mathcal{H}}^m$ be the regularized error. Let us recall the definition of uniform stability.

Definition D.1 *An algorithm $Z \to f_Z$ is said β uniformly stable if and only if $\forall n \geq 1$, $\forall 1 \leq i \leq n$, $\forall Z$ a realization of n i.i.d. copies of $(\underline{X}, \underline{Y})$,$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$ a Z independent realization of $(\underline{X}, \underline{Y})$, we have $|c(y, f_Z, x) - c(y, f_{Z^i}, x)| \leq \beta$.*

To prove the stability of a learning algorithm, it is common to make the following assumptions.

Assumption 3 *$\exists C_y > 0$ such that $|\underline{Y}| < C_y$ a.s.*

Assumption 4 *$\exists \kappa > 0$ such that $\sup_{x \in \mathcal{X}} k(x, x) < \kappa^2$*

Lemma D.2 *If Hypotheses 3 and 4 hold, then $\forall n \geq 1$, $\forall 1 \leq i \leq n$, $\forall Z$ a realization of n i.i.d. copies of $(\underline{X}, \underline{Y})$,$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$ a Z independent realization of $(\underline{X}, \underline{Y})$,*

$$|c(y, f_Z, x) - c(y, f_{Z^i}, x)| \leq C |f_Z(x) - f_{Z^i}(x)|,$$

with $C = 2 \left(C_y + \kappa \left(\frac{C_y^2}{\lambda} \right)^{\frac{1}{m}} \right)$.

Proof : Since $\mathcal{H}$ is a vector space, $0 \in \mathcal{H}$, and

$$\begin{aligned} \lambda \|f_Z\|^m &\leq \frac{1}{n} \sum_{i=1}^{n} (y_i - f_Z(x_i))^2 + \lambda \|f_Z\|_{\mathcal{H}}^m \\ &\leq \frac{1}{n} \sum_{k=1}^{n} \|y_k - 0\|^2 + \lambda \|0\|_{\mathcal{H}}^m \leq C_y^2, \end{aligned}$$

where we used the definition of f_Z as the minimum of (8) and Hypothesis 3. Using the reproducing property and Hypothesis 4, we deduce that

$$|f_Z(x)| \leq \sqrt{k(x, x)} \|f_Z\|_{\mathcal{H}} \leq \kappa \|f_Z\|_{\mathcal{H}} \leq \kappa \left(\frac{C_y^2}{\lambda} \right)^{\frac{1}{m}}.$$

The same reasoning holds for f_{Z^i}. Finally,

$$\begin{aligned} &|c(y, f_Z, x) - c(y, f_{Z^i}, x)| \\ &\quad = |(y - f_Z(x))^2 - (y - f_{Z^i}(x))^2| \\ &\quad \leq 2 \left(C_y + \kappa \left(\frac{C_y^2}{\lambda} \right)^{\frac{1}{m}} \right) |f_Z(x) - f_{Z^i}(x)|. \end{aligned}$$

□

The stability of our algorithm when $m \geq 2$ is established in the following theorem, whose proof is an extension of Theorem 22 in [4]. The original proof concerns the KRR case when $m = 2$. The beginning of our proof is similar to the original one; but starting from (17), the proof is modified to hold for $m \geq 2$, since the equalities used in [4] no longer holds when $m > 2$,. We use inequalities involving generalized Newton binomial theorem instead.

Theorem 3 *Under the assumptions 3 and 4, algorithm $Z \to f_Z$ defined in (8) is β stable $\forall m >= 2$ with*

$$\beta = C\kappa \left(2^{m-2}\frac{C\kappa}{\lambda n}\right)^{\frac{1}{m-1}}.$$

PROOF : Since c is convex with respect to f, we have $\forall 0 \leq t \leq 1$

$$\begin{aligned} &c(y, f_Z + t(f_{Z^i} - f_Z), x) - c(y, f_Z, x) \\ &\qquad \leq t\left(c(y, f_{Z^i}, x) - c(y, f_Z, x)\right). \end{aligned}$$

Then, by summing over all couples (x_k, y_k) in Z^i,

$$\begin{aligned} &R_e(f_Z + t(f_{Z^i} - f_Z), Z^i) - R_e(f_Z, Z^i) \\ &\qquad \leq t\left(R_e(f_{Z^i}, Z^i) - R_e(f_Z, Z^i)\right). \end{aligned} \tag{14}$$

By symmetry, (14) holds if Z and Z_i are permuted. By summing this symmetric equation and (14), we obtain

$$\begin{aligned} &R_e(f_Z + t(f_{Z^i} - f_Z), Z^i) - R_e(f_Z, Z^i) \\ &+ R_e(f_{Z^i} + t(f_Z - f_{Z^i}), Z^i) - R_e(f_{Z^i}, Z^i) \leq 0. \end{aligned} \tag{15}$$

Now, by definition of f_Z and f_{Z^i},

$$\begin{aligned} &R_r(f_Z, Z) - R_r(f_Z + t(f_{Z^i} - f_Z), Z) \\ &+ R_r(f_{Z^i}, Z^i) - R_r(f_{Z^i} + t(f_Z - f_{Z^i}), Z^i) \leq 0. \end{aligned} \tag{16}$$

By using (15) and (16) we get

$$\begin{aligned} &c(y_i, f_Z, x_i) - c(y_i, f_Z + t(f_{Z^i} - f_Z), x_i) \\ &\qquad + \lambda n\left(\|f_Z\|_{\mathcal{H}}^m - \|f_Z + t(f_{Z^i} - f_Z)\|_{\mathcal{H}}^m\right. \\ &\qquad \left. + \|f_{Z^i}\|_{\mathcal{H}}^m - \|f_{Z^i} + t(f_Z - f_{Z^i})\|_{\mathcal{H}}^m\right) \leq 0, \end{aligned} \tag{17}$$

This inequality holds $\forall t \in [0, 1]$. By choosing $t = 1/2$ in (17), we obtain that

$$\begin{aligned} &|c(y_i, f_Z, x_i) - c(y_i, f_Z + \frac{1}{2}(f_{Z^i} - f_Z), x_i)| \\ &\qquad \geq n\lambda\left(\|f_Z\|_{\mathcal{H}}^m - 2\left\|\frac{f_{Z^i} + f_Z}{2}\right\|_{\mathcal{H}}^m + \|f_{Z^i}\|_{\mathcal{H}}^m\right), \end{aligned} \tag{18}$$

Let $u = (f_Z + f_{Z^i})/2$ and $v = (f_Z - f_{Z^i})/2$. Then,

$$\begin{aligned} &\|u + v\|_{\mathcal{H}}^m + \|u - v\|_{\mathcal{H}}^m - 2\|u\|_{\mathcal{H}}^m - 2\|v\|_{\mathcal{H}}^m \\ &= \|f_Z\|_{\mathcal{H}}^m + \|f_{Z^i}\|_{\mathcal{H}}^m - 2\left\|\frac{f_{Z^i} + f_Z}{2}\right\|_{\mathcal{H}}^m - 2\left\|\frac{f_{Z^i} - f_Z}{2}\right\|_{\mathcal{H}}^m \\ &= \left(\|u\|_{\mathcal{H}}^2 + \|v\|_{\mathcal{H}}^2 + 2\langle u, v\rangle_{\mathcal{H}}\right)^{m/2} - 2\left(\|u\|_{\mathcal{H}}^2\right)^{m/2} \\ &+ \left(\|u\|_{\mathcal{H}}^2 + \|v\|_{\mathcal{H}}^2 - 2\langle u, v\rangle_{\mathcal{H}}\right)^{m/2} - 2\left(\|v\|_{\mathcal{H}}^2\right)^{m/2} \\ &\geq 2\left(\|u\|_{\mathcal{H}}^2 + \|v\|_{\mathcal{H}}^2\right)^{m/2} - 2\left(\|u\|_{\mathcal{H}}^2\right)^{m/2} - 2\left(\|v\|_{\mathcal{H}}^2\right)^{m/2} \\ &\geq 0, \end{aligned}$$

where in the last transition we used both Newton's generalized binomial theorem for the first inequality and the fact that $m/2 > 1$ for the second one. Hence, we have shown that

$$\|f_Z\|_{\mathcal{H}}^m - 2\left\|\frac{f_{Z^i} + f_Z}{2}\right\|_{\mathcal{H}}^m + \|f_{Z^i}\|_{\mathcal{H}}^m \geq 2\left\|\frac{f_{Z^i} - f_Z}{2}\right\|_{\mathcal{H}}^m. \tag{19}$$

Now, by combining (18) and (19), we obtain by using Lemma D.2,

$$\begin{aligned}
&\|f_Z - f_{Z^i}\|_{\mathcal{H}}^m \\
&\leq \frac{2^{m-1}}{\lambda n}\left(c(y_i, f_Z + \frac{1}{2}(f_{Z^i} - f_Z), x_i) - c(y_i, f_Z, x_i)\right) \\
&\leq 2^{m-2}\frac{C}{\lambda n}\|f_{Z^i}(x_i) - f_Z(x_i)\|_{\mathcal{Y}} \\
&\leq 2^{m-2}\frac{C\kappa}{\lambda n}\|f_{Z^i} - f_Z\|_{\mathcal{H}},
\end{aligned}$$

which gives that

$$\|f_Z - f_{Z^i}\|_{\mathcal{H}} \leq \left(2^{m-2}\frac{C\kappa}{\lambda n}\right)^{\frac{1}{m-1}}.$$

This implies that, $\forall(x, y)$ a realization of (X, Y),

$$\begin{aligned}
|c(y, f_Z, x) - c(y, f_{Z^i}, x)| &\leq C\|f_Z(x) - f_{Z^i}(x)\|_{\mathcal{Y}} \\
&\leq C\kappa\left(2^{m-2}\frac{C\kappa}{\lambda n}\right)^{\frac{1}{m-1}}.
\end{aligned}$$

□

For $1 < m < 2$, the problem (8) is well posed but the question whether the algorithm is stable or not in this case remains open. Future studies need to be conducted to further address this issue explicitly.

E Additional Experiments on M-RLSR

In this section, we conduct experiments on synthetic and real-world datasets to evaluate the efficiency of the proposed algorithm. We use the following real-world datasets extracted from the UCI repository[3]: Concrete Compressive Strength (1030 instances, 9 attributes), Concrete Slump Test (103 instances, 10 attributes), Yacht Hydrodynamics (308 instances, 7 attributes), Wine Quality (4898 instances, 12 attributes), Energy Efficiency (768 instances, 8 attributes), Housing (506 instances, 14 attributes) and Parkinsons Telemonitoring (5875 instances, 26 attributes). Additionally, we also use a synthetic dataset (2000 instances, 10 attributes) described in [19]. In this dataset, inputs $(x_1, ..., x_{10})$ are generated independently and uniformly over $[0, 1]$ and outputs are computed from $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \mathcal{N}(0, 1)$. In all our experiments, we use a Gaussian kernel $k_\mu(x, x') = \exp(-\|x - x'\|_2^2/\mu)$ with $\mu = \frac{1}{n^2}\sum_{i,j}\|x_i - x_j\|_2^2$, and the scaled root mean square error (RMSE), defined by $\frac{1}{\max y_i}\sqrt{\frac{1}{n}\sum_i (y_i - f(x_i))^2}$, as evaluation measure.

E.1 Speed of Convergence

We compare here the convergence speed of M-RLSR with $m \leq 1$ and KRR on Concrete compressive strength, Yacht Hydrodynamics, Housing, and Synthetic datasets. As before, each dataset is randomly split into two parts (70% for learning and 30% for testing). The parameters m and λ are selected using cross-validation: we first fix λ to 1 and choose m over a grid ranging from 0.1 to 1, then λ is set by cross-validation when m is fixed. For KRR, λ_2 is computed from λ and m using Proposition 5.1.

Figure 3 shows the mean of RMSE over ten run for the four datasets with M-RLSR and KRR when varying the number of examples of training data from 10% to 100% with a step size of 5%. In this figure, we can see that M-RLSR with $m < 1$ can improve the speed of convergence of KRR.

[3] `http://archive.ics.uci.edu/ml/datasets`.

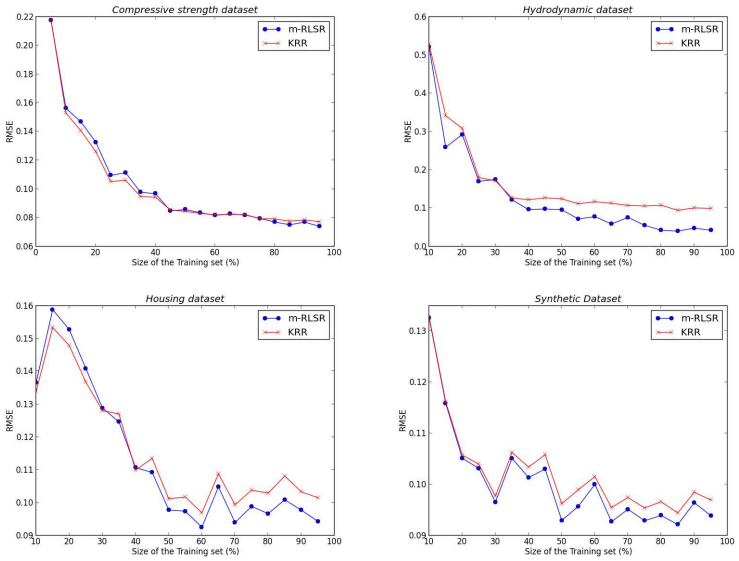

Figure 3: RMSE curve of M-RLSR (blue) and KRR (red) algorithms as a function of the dataset size. (top left) Concrete compressive strength ($m = 0.1$). (top right) Yacht Hydrodynamics ($m = 0.5$). (bottom left) Housing ($m = 0.4$). (bottom right) Synthetic ($m = 0.1$).

This confirms the theoretical expectation for this situation [7], that is a regularization exponent that grows significantly slower than the standard quadratic growth in the RKHS norm can lead to better convergence behavior.

E.1.1 Prediction Accuracy

We evaluate the prediction accuracy of the M-RLSR algorithm using the datasets described above and compare it to KRR. For each dataset we proceed as follows: the dataset is split randomly into two parts (70% for training and 30% for testing), we set $\lambda = 1$, and we select m using cross-validation in a grid varying from 0.1 to 2.9 with a step-size of 0.1. The value of m with the least mean RMSE over ten run is selected.Then, with m now fixed, λ is chosen by a ten-fold cross validation in a logarithmic grid of 7 values, ranging from 10^{-5} to 10^{2}. Likewise, λ_2 for KRR is chosen by 10-fold cross-validation on a larger logarithmic grid of 25 equally spaced values between 10^{-7} and 10^{3}.

RMSE and standard deviation (STD) results for M-RLSR and KRR are reported in Table 1. It is important to note that the double cross-validation on m and λ for M-RLSR, and the cross-validation on the greater grid for the KRR takes a similar amount of time. Table 1 shows that the m-power RLSR algorithm is capable of achieving a good performance results when $m < 2$. Note that the difference between the performance of the two algorithms M-RLSR and KRR decreases as the grid of λ becomes larger, but in practice we are limited by computational reasons.

Table 1: Performance (RMSE and STD) of m-power RLSR (M-RLSR) and KRR algorithms on synthetic and UCI datasets. m is chosen by cross-validation on a grid ranging from 0.1 to 2.9 with a step-size of 0.1.

	KRR		M-RLSR		
Dataset	RMSE	STD	m	RMSE	STD
Compressive	8.04e-2	3.00e-3	1.6	7.31e-2	3.67e-3
Slump	3.60e-2	5.62e-3	1.1	3.52e-2	6.49e-3
Yacht Hydro	0.165	1.13e-2	0.1	1.56e-2	7.53e-3
Wine	8.65e-2	6.18e-3	1.3	8.17e-2	6.07e-3
Energy	4.12e-2	1.79e-3	1.1	3.79e-2	2.87e-3
Housing	10.6e-2	7.98e-3	1.3	7.26e-2	9.92e-3
Parkinson	8.05e-2	4.51e-3	0.3	5.56e-2	3.29e-3
Synthetic	3.19e-2	1.56e-3	0.4	1.26e-2	5.85e-4